VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zpracování výukových programů pro předmět Základy počítačové grafiky

Working out the teaching computer programs for the course „Principles of Computer Graphics“

Rok 2008/2009

Pavlína Hamerská

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě …………………………. ………………………………

Pavlína Hamerská

Poděkování

Chtěla bych poděkovat panu RNDr. Arnoštu Šarmanovi, který je mým vedoucím bakalářské práce, za poskytnutí hodnotných rad a informací k vypracování této bakalářské práce. Dále bych chtěla poděkovat svým blízkým za velkou podporu a možnost se práci věnovat.

Abstrakt

Cílem bakalářské práce je seznámit se s problematikou realizace řešení geometrických úloh v prostředí počítačové grafiky a seznámit se s SW prostředím Macromedia Flash.

Klíčové slova

Cyklické křivky, cykloida, epicykloidy, hypocykloidy,ekvidistantní křivky, evolventy, evoluty, spirály, konchoidální křivky.

Abstract

The aim of this bachelor study is to introduce a solution for geometry theories using animations as visual demonstration designed by animation and programming environment Macromedia Flash.

Keyword

Cyclical curves, cycloids, epicycloids, hypocycloids, equidistant curves, involutes, evolutes, spirals, conchoidal curves.

Obsah

1. Úvod

1.1. Popis jednotlivých kapitol

V první části této bakalářské práce se seznámíme s historií křivek a s historií Macromedia Flash.

V druhé části nás čeká hlubší seznámení s jednotlivými křivkami jako je např. jejich definice apod.

Ve třetí části se budeme zabývat praktickou částí této práce a to realizací řešení geometrických úloh v prostředí počítačové grafiky.

A na závěr si shrneme jednotlivé části a uzavřeme tuto práci vypsáním použité literatury.

1.2. Historie křivek

Křivky se nacházejí všude kolem nás a mi o tom ani nevíme. V přírodě najdeme plno křivek jako je třeba krk labutě, který je ve tvaru S nebo sílu větru či vody.

Každý ví, co pojem křivka znamená, ale málo kdo umí definovat přesný význam tohoto slova. Zkusme si něco říct o tom, jak to s křivkou bylo od pradávna, kdy se její problematikou zabývalo mnoho geometrů a matematiků.

Když se podíváme hluboko do historie, kdy se lidé začali o křivky zajímat tak přesné datum nenajdeme. Zájem o křivky začal v době, kdy byl spjat s každodenní činnosti člověka například k dělení pozemků, k vytyčování obřadních míst apod.

Přibližně v 6.stoleti před n. l. se ve starověkém Řecku dostává do popředí zájmu geometrie jako vhodný nástroj k popisu světa a jevů v něm. Mezi významné matematiky a geometry patřil i Euklides, který ve svém spise Základy vykládá geometrii jako axiomaticky budovaný systém. Tento velký zlom sebou přinesl nový přístup ke křivkám, kružnicím, přímkám, geometrickým objektům. Zájem o jejich vlastnosti a vzájemné vztahy byly předmětem zájmů mnoha středověkých geometrů.

Po rozpadu antického světa na území Evropy nastává útlum zájmu o křivky, geometrii a vědu vůbec.

Na konci středověku se v Evropě pomalinku začíná objevovat některé křivky jako součást náročných gotických staveb. Konstrukce těchto nádherných architektonických prvků se neobešly bez jistých geometrických znalostí.

V roce 1890 italský matematik Peano ukázal, že existuje spojité zobrazení úsečky na čtverec, jinými slovy, že dráha hmotného bodu, který se spojitě pohybuje, může vyplňovat cely čtverec. Toto spojité zobrazeni, ale není vzájemně jednoznačné, tj. pohybující se bod projde některými body čtverce několikrát. Přesnější definici křivky podal rakouský matematik Menger v roce 1921 a nezávisle na něm téměř současně sovětský matematik Urysohn. Jejich definici lze uvést asi takto: křivka je kontinuum, jehož libovolné dva body se dají oddělit množinou neobsahující již žádné kontinuum.

Vývoj pojmu křivka tímto zdaleka neskončil, charakteristika tohoto pojmu se neustále zpřesňuje. V současné době se definice křivky provádí pomocí tzv. variet.

1.3. Historie Macromedia Flash

Macromeida Flash vznikl v roce 1994, kdy jeho jméno nebylo ještě Flash, ale SmartSketch. SmartSketch byl založen na programovacím jazyce Java. Bohužel Java nevyhovovala očekávaným požadavkům, a proto se muselo od ní ustoupit.

V roce 1995 se objevuje prohlížeč, který podporuje zásuvné moduly typu Plug-in a tak je SmartSketch přejmenován na FutureSplash Animator.

V roce 1996 Macromedia kupuje FutureSplash Animator a vzniká tak Macromedia Flash 1.0. Tato verze neobsahuje skriptování ActrionScript, ale dává naději na úspěšný vývoj webových animací.

Následující verze Macromedia Flash 2.0 umožňuje základní skriptovou manipulaci s přehráváním animací a objevují se prvky jako tlačítko a grafika.

Třetí verze přichází s ozvučením animací a s tím spojené příkazy, dále pak s příkazy typu fscommand a možnost skriptově načítat SWF animace.

Významnou verzí je verze 4, kdy celý ActionScript je přepracován a vzniká velké množství příkazů, příkazy pro movieclip, podmínky, smyčky, nové operátory, načítání proměnných ze souboru aj.

Mezi další významnou verzi patří verze 5. V této verzi vznikají objekty se svými metodami a vlastnostmi. Je možné vytvářet vlastní funkce, dále vznikají komponenty a většina příkazů je přeorientována na objekty. Vzniká přehlednější dot syntaxe a podporována je komunikace se serverem pomocí XML Socket.

Verze 6 je rozšířena o objekty a metody. Vznikají UI komponenty a také vzniká spolupráce s videem. V této verzi dochází k podporování obousměrnému streamovému přenosu zvuku a videa pomocí kamer a mikrofonů. Dále je zde vyvinut nový komunikační protokol RTMP a Server-side ActionScript pro komunikaci se serverovými službami a serverový balík Flash Communication Server MX.

V sedmé verzi dochází k rozdělení produktu na dvě podoby: MX 2004 a MX 2004 Pro. Obě verze nabízí vylepšené rozhraní, nové efekty, vylepšenou časovou osu, rozšíření kompatibility importovaných objektů (PDF, EPS, …), konečně také panel „historie“, šablony pro vytvoření složitých Motion a Shape Tweenů, vylepšené trasování bitmap a ActionScript 2.0, který se vyznačuje větší přímočarostí a robustností.

Verze Professional nabízí podporu externího ActionScriptu a lepší zpracování videových souborů, podporu zvuků ve formátu MIDI pro mobilní zařízení, vylepšené formuláře a jednoduché vytváření slidů a prezentací ve speciálním módu.

V roce 2005 Adobe Inc. kupuje společnost Macromedia s celým portfoliem produktů a následuje vydání nového produktu Flash 8.0. Tato verze se vyznačuje pokročilou prací s videem včetně alpha kanálu, blend filtry, možností cacheovat vektory jako rastry, ClearType™ antialiasingem písma, vytvářením multiplatformních animací a spoustou dalších vylepšení v AS2.0

2. Křivky
2.1. Cyklické křivky

Cyklickou křivku neboli trochoidu můžeme definovat jako množinu bodů roviny, kterými prochází body křivky k, která se pohybuje po pevné křivce p. Vzniklé křivce říkáme kotálnice.

Mezi cyklické křivky patří:

- Cykloidy
- Epicykloidy
- Hypocykloidy

Mezi cyklické křivky také může patřit např. evolventa kružnice.

2.1.1. Cykloida

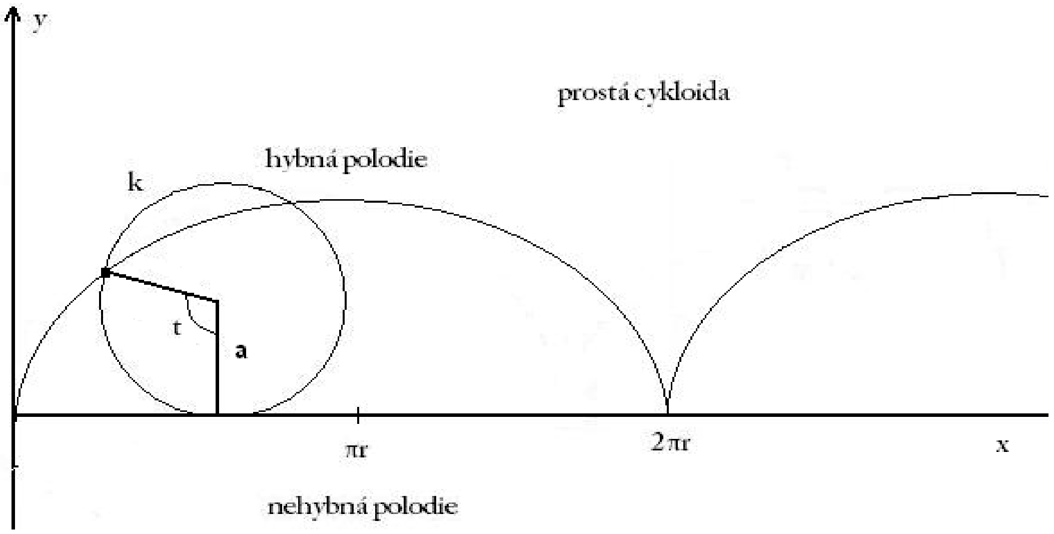
Cykloida je cyklická křivka, která je vytvořena bodem pevně spojeným s kružnicí, která se kutálí po přímce.

Rozdělujeme ji na ***cykloidu prostou s body vratu, zkrácenou s inflexními body (nebo bez nich) a prodlouženou s uzlovými body***.

Nejdříve si něco povíme o cykloidě prosté.

Cykloida prostá

Kružnice kutálející se po přímce opisuje bod pevně ležící na jejím obvodu, tomuto opisování říkáme ***cykloida prostá.***



Obr. 1 ***Prostá*** cykloida

Poznámka:

Nehybná i hybná polodie je geometrické místo bodů, kdy u nehybné polodie jsou body, které jsou okamžitým středem otáčení pro příslušnou polohu pohybu a pro hybnou polodii jsou to body, které se během pohybu stanou okamžitými středy otáčení.

Vždy se obě polodie dotýkají v bodě, která je okamžitým středem otáčení.

Každou cykloidu lze vyjádřit parametrickou rovnicí:

$$x = a(t - \sin t)$$

$$y = a(1 - \cos t)$$

Kde ***a*** představuje poloměr kružnice a ***t*** je parametr odpovídající délce oblouku kotálející se kružnice.

Prostou cykloidu lze vyjádřit i v explicitním tvaru:

$$x = a \arccos \frac{a-y}{a} - \sqrt{y(2a-y)}$$

pro $y \in \langle 0, 2a \rangle$, neboli

$$x = a\left(2\pi - \arccos \frac{a-y}{a}\right) + \sqrt{y(2a-y)}$$

pro $y \in \langle 0,2a \rangle$

Cykloida má periodu $2\pi a$.

Délka oblouku dané větve prosté cykloidy od vrcholu do ***bodu [x, y]*** je

$$s = 4a\left(1 - \cos \frac{t}{2}\right)$$

Dosazením periody získáme pro délku jedné větve prosté cykloidy výraz

$$s = 8a$$

Poloměr křivosti v bodě různém od hrotu prosté cykloidy je

$$r = \left|4a \sin \frac{t}{2}\right|$$

Poloměr první křivosti ve vrcholu je

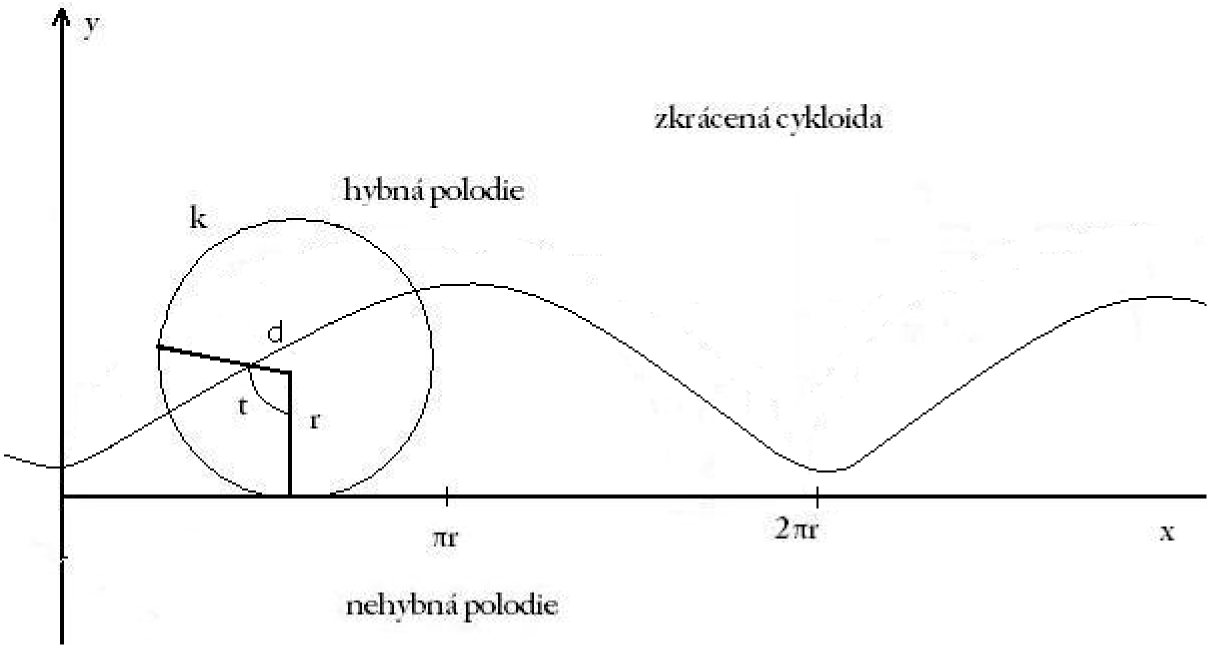
$$r = 4a$$

Obsah plochy ohraničené jednou větví prosté cykloidy je

$$S = 3\pi a^2$$

Cykloida zkrácená

Sledujeme-li bod kutálející kružnice, který leží na poloměru ve vzdálenosti ***d* < *r*** od středu kružnice, uvidíme ***cykloidu zkrácenou***.



Obr. 2 ***Zkrácená*** cykloida

Parametrická rovnice zkrácené cykloidy

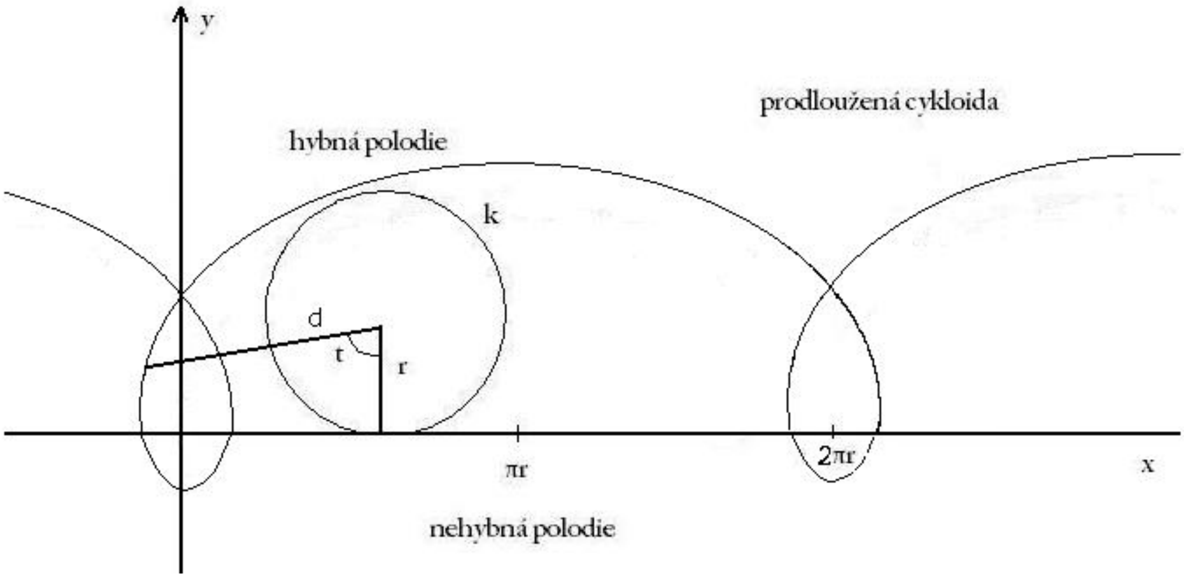
$$x = rt - d\sin t$$

$$y = r - d\cos t$$

$t \in \langle 0; 2\pi \rangle$

Cykloida prodloužená

Je úplně stejná jako cykloida zkrácená jenom se liší ve vzdálenosti poloměru a to ***d*** **>** ***r*.**



Obr. 3 ***Prodloužená*** cykloida

Parametrická rovnice prodloužené cykloidy je stejná jako u zkrácené cykloidy

$$x = rt - d \sin t$$

$$y = r - d \cos t$$

$t \in \langle 0; 2\pi \rangle$

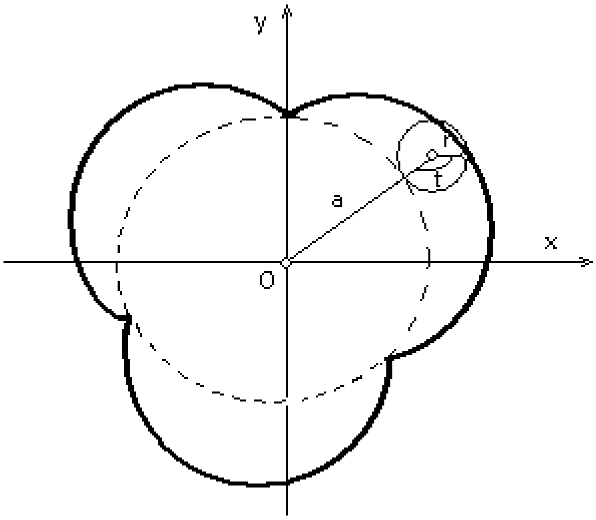
2.1.2. Epicykloida

Vyměníme-li přímku, po které se kutálí kružnice, za nehybnou kružnici, vznikne nám s cykloidy ***epicykloida.***

Epicykloidu také rozdělujeme na ***epicykloidu prostou, zkrácenou a prodlouženou.***

<u>Epicykloida prostá</u>

Pokud necháme pohyblivou kružnici o poloměru ***r*** válet po vnější straně nehybné kružnice o poloměru ***a*** vznikne epicykloida ***prostá***.



Obr. 3 ***Prostá*** epicykloida

Epicyklickou křivku lze vyjádřit parametrickou rovnicí a to

$$x = (a + r)\cos t - r\cos\left(\frac{a + r}{r}\right)t$$

$$y = (a + r)\sin t - r\sin\left(\frac{a + r}{r}\right)t$$

Je- li poměr:

$\frac{a}{r} = m$ celé číslo, potom je epicykloida prostá uzavřenou křivkou s ***m*** větvemi, které vznikají při jednom oběhu hybné kružnice kolem nehybné kružnice.

$\frac{a}{r} = \frac{p}{q} = m$ racionální číslo v základním tvaru $\frac{p}{q}$, potom prostá epicykloida je uzavřenou křivkou s ***p*** větvemi, které vznikají při ***q*** oběhu hybné kružnice kolem nehybné kružnice.

$\frac{a}{r} = m$ iracionální číslo, potom prostá epicykloida není uzavřenou křivkou a obsahuje nekonečně mnoho větví.

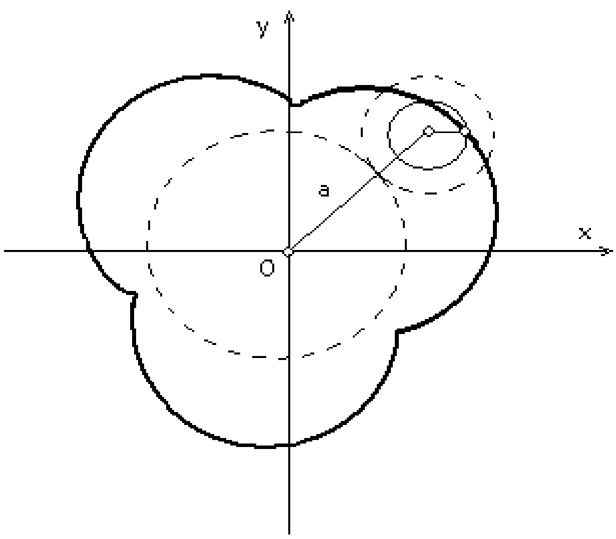
Epicykloida zkrácená a prodloužená

Vztahy parametrické rovnice epicykloidy zkrácené a prodloužené jsou

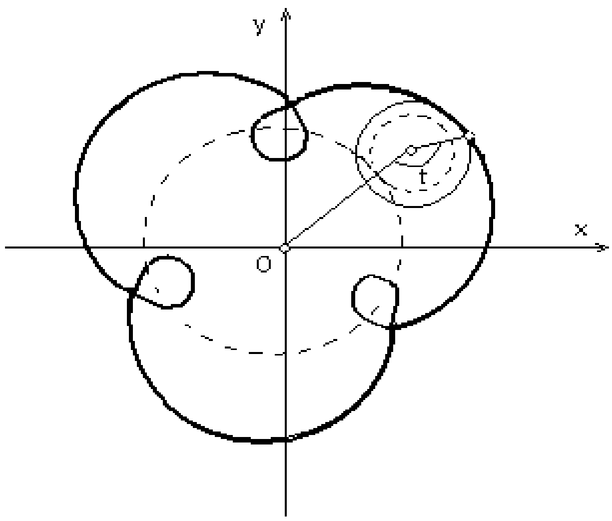
$$x = (a + r)\cos t - d\cos\left(\frac{a + r}{r}\right)t$$

$$y = (a + r)\sin t - d\sin\left(\frac{a + r}{r}\right)t$$

kde ***r*** je poloměr pohyblivé kružnice, ***d*** je poloměr odvalující se kružnice, která je na obr. 4 a obr. 5 vyznačená plnou čárou, ***a*** je poloměr nehybné kružnice, kterou máme vyznačenou čárkovaně. Je-li ***d*** < ***r*** potom se jedná o ***epicykloidu zkrácenou***, ale pokud je ***d*** > ***r*** jedná se o ***epicykloidu prodlouže-nou***.



Obr. 4 ***Zkrácená*** epicykloida



Obr. 5 ***Prodloužená*** epicykloida

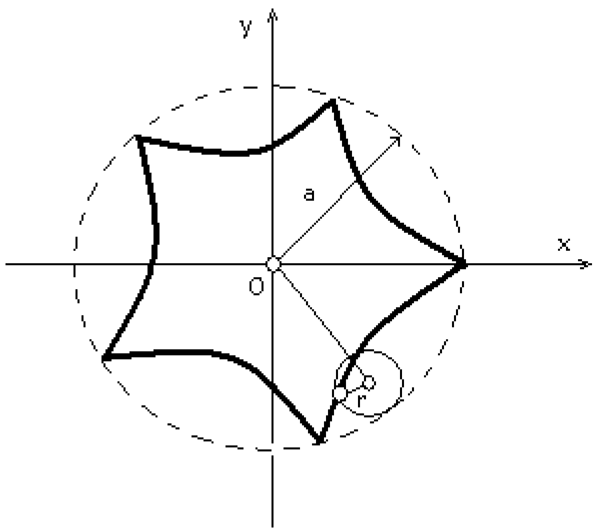
2.1.3. Hypocykloida

Hypocykloidu tvoří bod, který je pevně spojený s kutálející kružnicí po vnitřní straně nehybné kružnice.

Hypocykloida je ***zkrácená, prodloužená a prostá***.

Hypocykloida prostá

Pokud necháme pohyblivou kružnici o poloměru ***r*** válet po vnitřní straně nehybné kružnice o poloměru ***a*** vznikne ***hypocykloida prostá***.



Obr. 6 ***Prostá*** hypocykloida

Parametrická rovnice pro hypocykloidu prostou je

$$x = (a - r)\cos t + r\cos \left(\frac{a-r}{r}\right)t$$

$$y = (a - r)\sin t - r\sin \left(\frac{a-r}{r}\right)t$$

Je- li poměr:

$\frac{a}{r} = m$ celé číslo, potom je hypocykloida prostá uzavřenou křivkou s ***m*** větvemi, které vznikají při jednom oběhu hybné kružnice kolem nehybné kružnice.

$\frac{a}{r} = \frac{p}{q} = m$ racionální číslo v základním tvaru $\frac{p}{q}$, potom prostá hypocykloida je uzavřenou křivkou s ***p*** větvemi, které vznikají při ***q*** oběhu hybné kružnice kolem nehybné kružnice.

$\frac{a}{r} = m$ iracionální číslo, potom prostá hypocykloida není uzavřenou křivkou a obsahuje nekonečně mnoho větví.

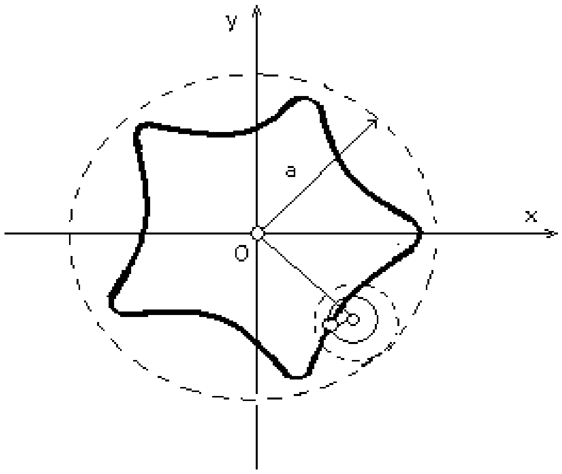
Hypocykloida zkrácená a prodloužená

Parametrická rovnice pro hypocykloidu zkrácenou a hypocykloidu prodlouženou je dána vztahy

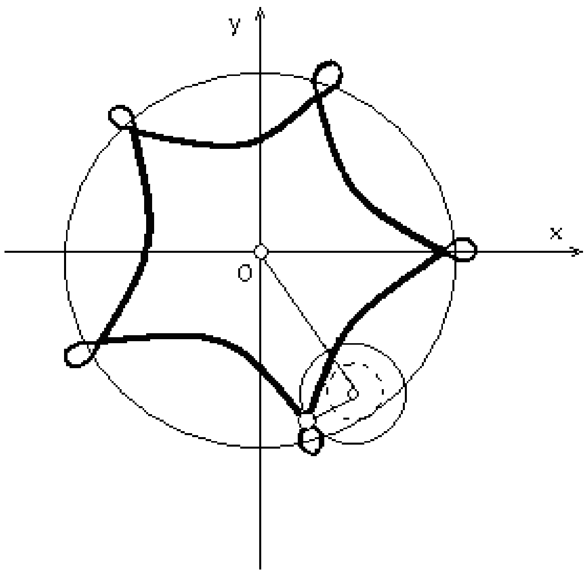
$$x = (a - r)\cos t + d\cos \left(\frac{a-r}{r}\right)t$$

$$y = (a - r)\sin t - d\sin \left(\frac{a-r}{r}\right)t$$

kde ***r*** je poloměr pohyblivé kružnice, která je znázorněna čárkovaně, ***d*** je poloměr odvalující se kružnice, která je na obr. 7 a obr. 8 vyznačená plnou čárou, ***a*** je poloměr nehybné kružnice, kterou máme vyznačenou čárkovaně. Je-li ***d*** < ***r*** potom se jedná o ***hypocykloidu zkrácenou***, ale pokud je ***d*** > ***r*** jedná se o ***hypocykloidu prodlouženou***.



Obr. 7 ***Zkrácená*** hypocykloida



Obr. 8 ***Prodloužená*** hypocykloida

2.2. Ekvidistantní křivky

Ekvidistantní křivky neboli rovinné křivky chápeme jako množinu bodů v rovině. Každá rovinná křivka má nekonečně mnoho rovnoběžných křivek pro různé hodnoty. Výjimkou jsou prostorové křivky, u kterých rovnoběžná křivka nemusí existovat. Rovinné křivky můžeme popsat různými způsoby, jako jsou

- Explicitní rovnicí

  $y = f(x)$

- Implicitní rovnicí

  $g(x, y) = 0$

- Parametrickou rovnicí

  $x = \varphi(t)$

  $y = \psi(t) \qquad t \in \langle a, b \rangle$

Pokud rovinnou křivku vyjádříme parametrickou rovnicí, potom je možné uvedenou rovnici vyjádřit tak to

$$\bar{x} = \varphi(t) + c\frac{\frac{d\psi}{dt}}{\sqrt{\left(\frac{d\varphi}{dt}\right)^2 + \left(\frac{d\psi}{dt}\right)^2}}$$

$$\bar{y} = \varphi(t) + c\frac{\frac{d\psi}{dt}}{\sqrt{\left(\frac{d\varphi}{dt}\right)^2 + \left(\frac{d\psi}{dt}\right)^2}}$$

kde $\bar{\boldsymbol{x}}$, $\bar{\boldsymbol{y}}$ vyjadřují souřadnice rovnoběžných křivek a **c** znázorňuje libovolnou reálnou konstantu. Jestliže je rovinná křivka rovnoběžná s další křivkou pak se nazývá ***ekvidistanta*** neboli ***ekvidistantní křivka***. Rovnici ekvidistanty k přímce ***F (x, y) = 0 vyjádříme*** tak, že vyloučíme proměnné ***x, y*** z rovnic

$$Y - y = -\frac{dy}{dx}(X - x)$$

$$(X - x)^2 + (Y - y)^2 = c^2$$

X a Y jsou souřadnice bodů ekvidistanty.

2.2.1. Evolventy

Když jsme si na začátku povídali o cykloidě, říkali jsme si, že je tvořena pohybem kružnice po pevné přímce. Teď se budeme zabývat pohybem přímky po nehybné kružnici. Tomuto pohybu říkáme ***evolventní pohyb***. ***Evolventa*** je přímka pevně spojená s bodem valící se po nehybné kružnici.

Evolventu sestavíme tak, že obvod dané kružnice rozdělíme na určitý počet stejných dílů, zrektifikujeme oblouk příslušný jednomu dílu a pak na příslušné tečny ke kružnici v dělících bodech naneseme příslušný počet délek zrektifikovaných oblouků.

Také evolventa je ***prostá, zkrácená a prodloužená***.

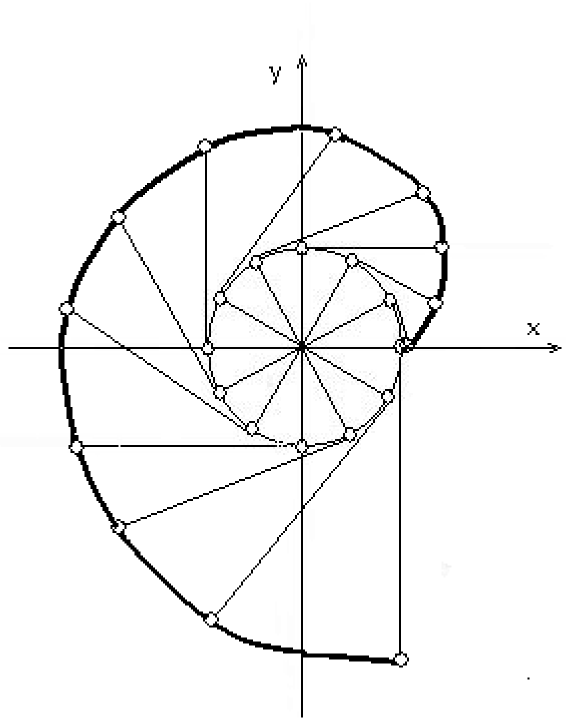
Evolventa prostá

Parametrická rovnice pro evolventu prostou je

$$x = (r + d)\cos t + rt\sin t$$

$$y = (r + d)\sin t - rt\cos t$$

kde ***r*** je poloměr, který máme znázorněný plnou čárou. Pokud je ***d = 0*** je ***evolventa prostá***.



Obr. 9 ***Prostá*** evolventa

Evolventa zkrácená a prodloužená

Parametrická rovnice pro evolventu zkrácenou a prodlouženou

$$x = (r + d)\cos t + rt\sin t$$

$$y = (r + d)\sin t - rt\cos t$$

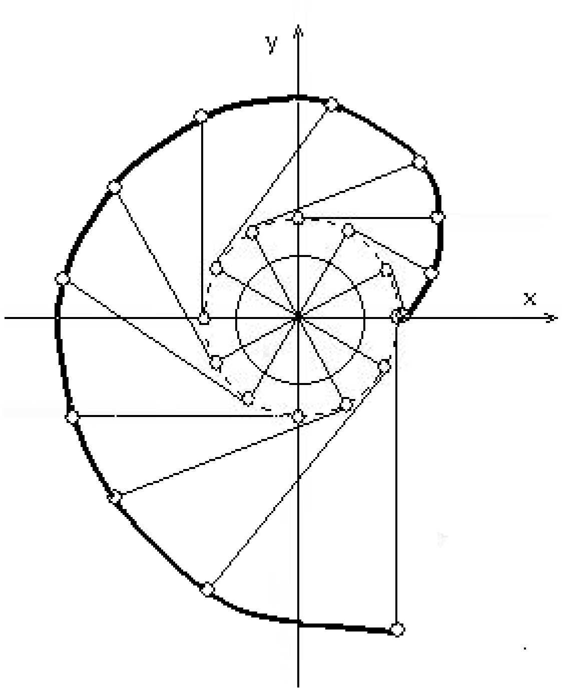
kde ***r*** je poloměr, který máme znázorněný plnou čárou. Pokud je ***d* > *0*** je ***evolventa zkrácená*** a pro

d* < *0 je ***evolventa prodloužená***.

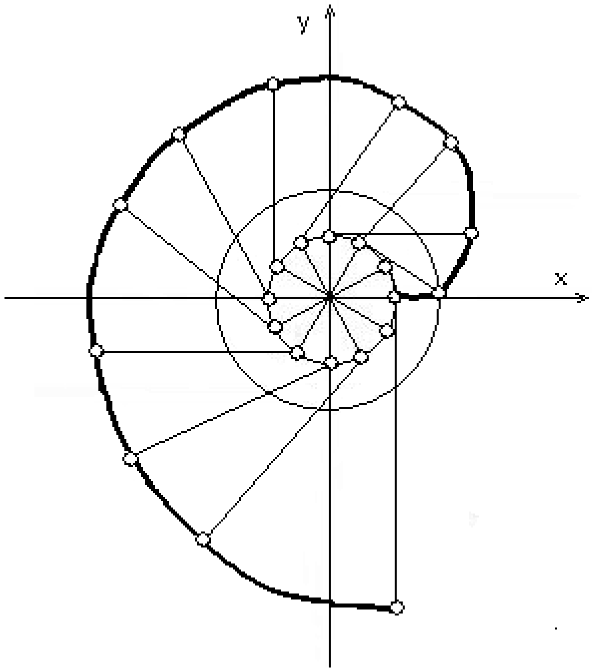
Pokud ***d* = - *r*** potom je evolventa prodloužená, také se jí říká Archimédova spirála, která má parametrickou rovnici

$$x = rt\sin t$$

$$y = -rt\cos t$$



Obr. 10 ***Zkrácená*** evolventa



Obr. 11 ***Prodloužená*** evolventa

2.2.2. Evoluta

Evoluta je křivka, která je tvořena středy křivostí bodů dané křivky. Také bychom mohli říct, že ***evoluta*** je opakem evolventy. Každá evoluta má k sobě nekonečně mnoho evolvent.

Pokud je evoluta ***k*** určena rovnicí $\boldsymbol{r} = \boldsymbol{r}(\boldsymbol{s})$, pak je evolventa ***l*** dána rovnicí

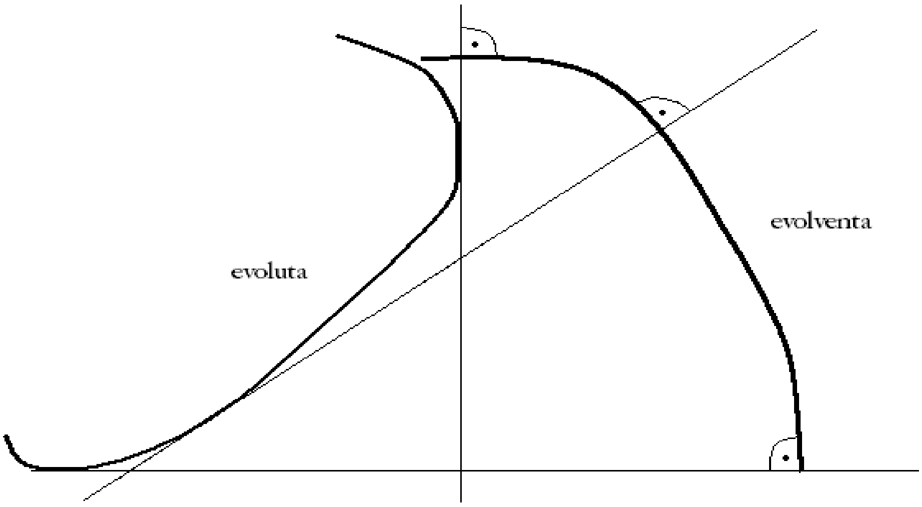
$$\bar{r} = r(s) - (s + c)t,$$

kde ***c*** znázorňuje libovolnou konstantu a ***t*** je tečný vektor evoluty.

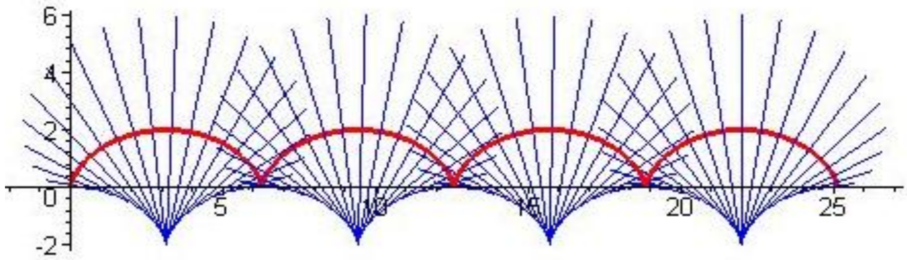
Pokud je evolventa ***l*** určena rovnicí $\bar{\boldsymbol{r}} = \bar{\boldsymbol{r}}(\boldsymbol{s})$, potom je evoluta ***k*** dána rovnicí

$$r = \bar{r}(s) + \bar{r}_1 \left[\bar{n} + \bar{b}\cot\left(\int k_2 ds + c\right)\right]$$

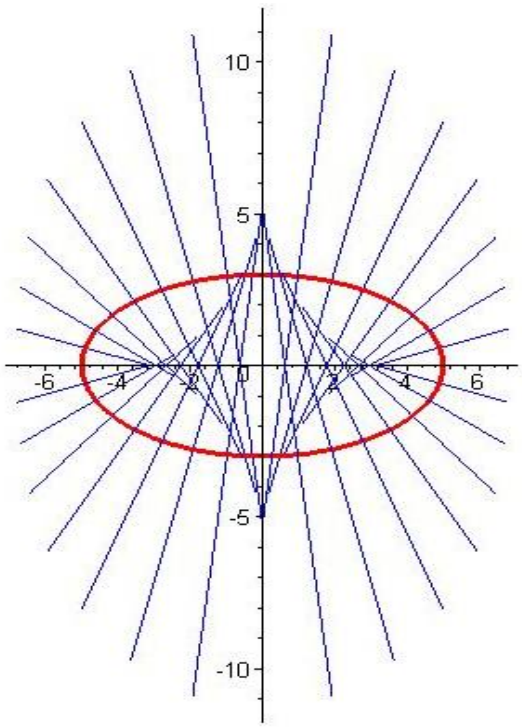
c znázorňuje libovolnou konstantu, $\bar{\boldsymbol{n}}$ je vektor hlavní normály, $\bar{\boldsymbol{b}}$ je vektor binormály a $\overline{\boldsymbol{r}_1}$ je poloměr křivosti evolventy.



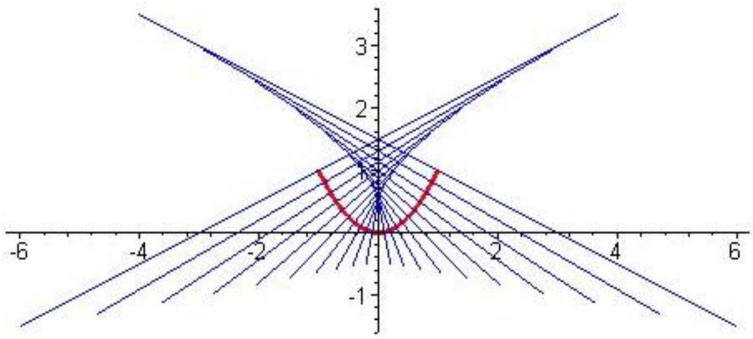
Obr. 12 Vztah evolventy s evolutou



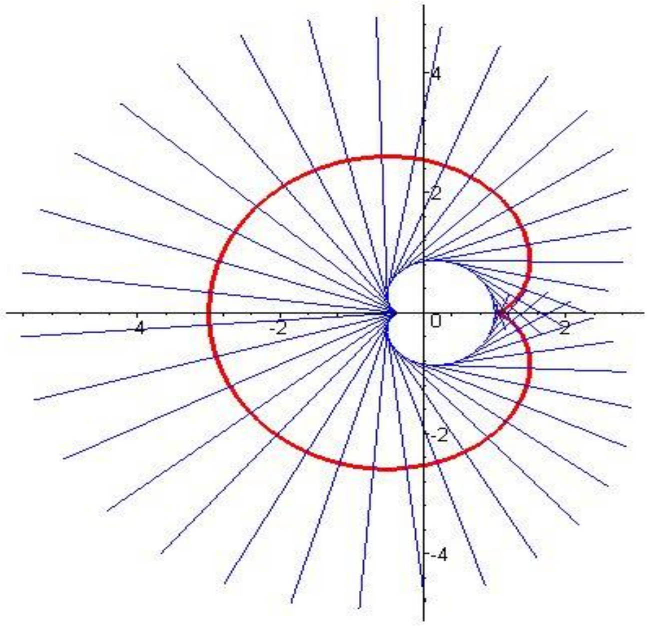
Obr. 13 Evoluta cykloidy



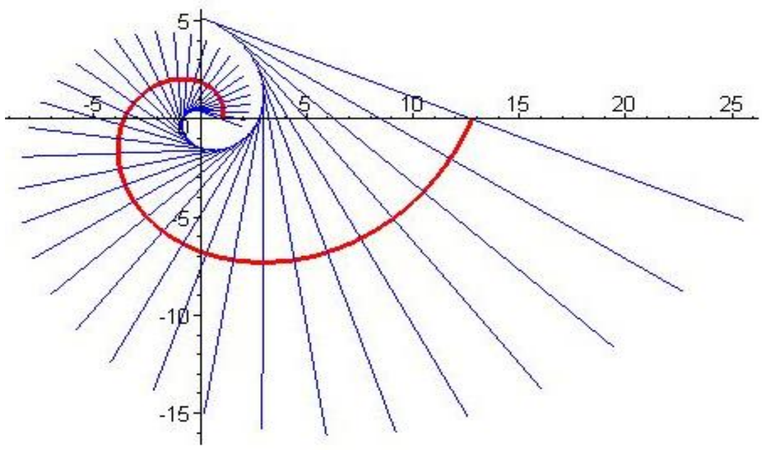
Obr. 14 Evoluta elipsy



Obr. 15 Evoluta paraboly



Obr. 16 Evoluta kardioidy



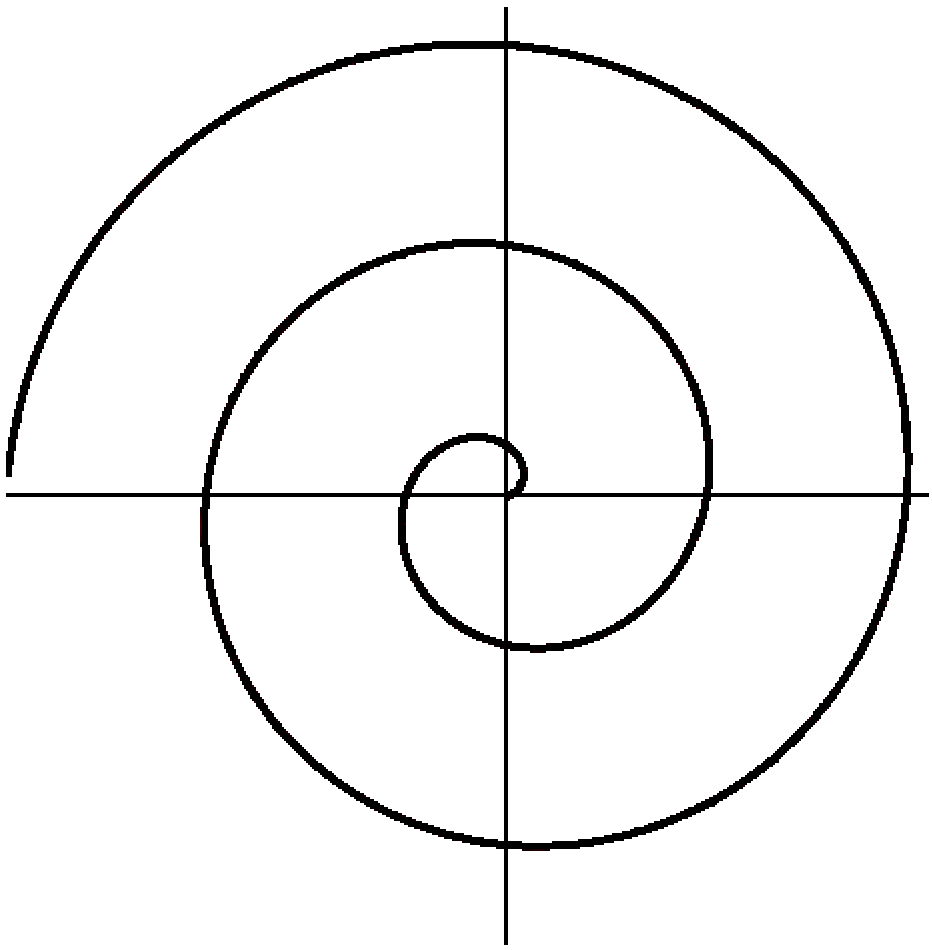
Obr. 17 Evoluta logaritmické spirály

2.3. *Spirála*

Spirála patří mezi rovinné křivky, které představují dráhu bodů pohybujících se po přímce podle daného pravidla, mezitím se přímka otáčí konstantní rychlostí kolem pevného bodu.

Budeme mluvit o těchto spirálách:

- Archimédova spirála
- Logaritmická spirála
- Hyperbolická spirála



Obr. 13 Spirála

2.3.1. *Archimédova spirála*

Jestliže se bod rovnoměrně pohybuje po přímce, potom je jeho vzdálenost úměrná úhlu tj.

$$\rho = k\alpha,$$

$\alpha \in R$

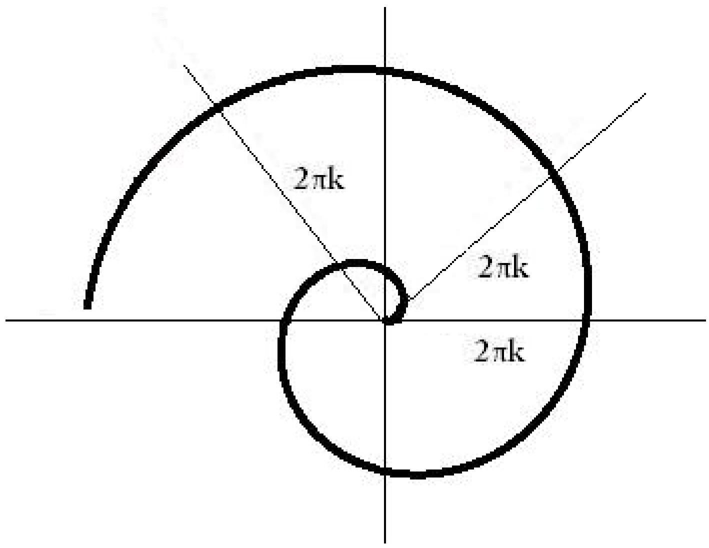
kde $\boldsymbol{k > 0}$ je koeficient úměrnosti.

Pokud chceme znát délku oblouku Archimédovy spirály můžeme ji určit ze vztahu

$$s = \frac{k}{2}\left(\alpha\sqrt{\alpha^2+1} \;\; + \operatorname{arcsinh}\alpha\right)$$

Pro výpočet poloměru křivosti Archimédovy spirály použijeme

$$R = \frac{(k^2+\rho^2)^{\frac{3}{2}}}{2k^2+\rho^2} = \frac{k(\alpha^2+1)^{\frac{3}{2}}}{\alpha^2+2}$$



Obr. 13 Archimédova spirála

2.3.2. Logaritmická spirála

Bod se pohybuje po dráze, která urazí stejné časové úseky a tak tvoří geometrickou posloupnost. Spirála protíná všechny přímky vycházející z počátku pod stejným úhlem γ.

Logaritmická spirála má v polárních souřadnicích podobu

$$r = ae^{b\varphi},$$

a, b > ***0*** vyjadřují konstantní parametry, ***r*** je poloměr (vektor spojující pól spirály a bod ležící na spirále).

Kdybychom tuto rovnici zlogaritmovali, získáme

$$ln\frac{r}{a} = b\varphi$$

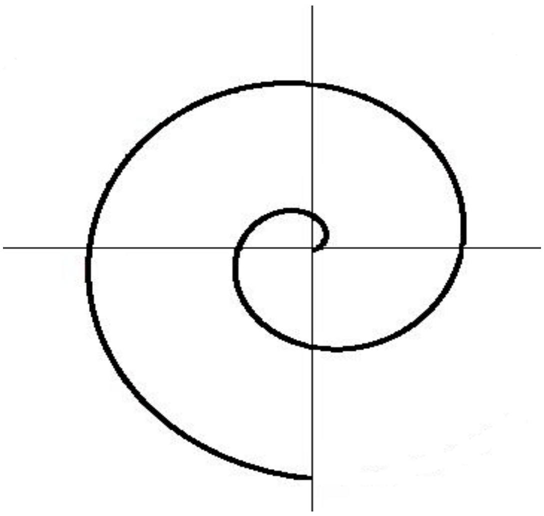
Díky tomuto vztahu vznikl název ***logaritmická spirála***.

Pokud chceme znát délku oblouku logaritmické spirály mezi dvěma body ve vzdálenosti poloměru ***r1*** a poloměru ***r2*** použijeme vztah

$$s = \frac{1}{b}|r_2 - r_1|\sqrt{1 + b^2} = |r_2 - r_1|\cos\gamma$$

Pro výpočet poloměru křivosti Logaritmické spirály použijeme

$$R = r\sqrt{1 + b^2}$$



Obr. 14 Logaritmická spirála

2.3.3. Hyperbolická spirála

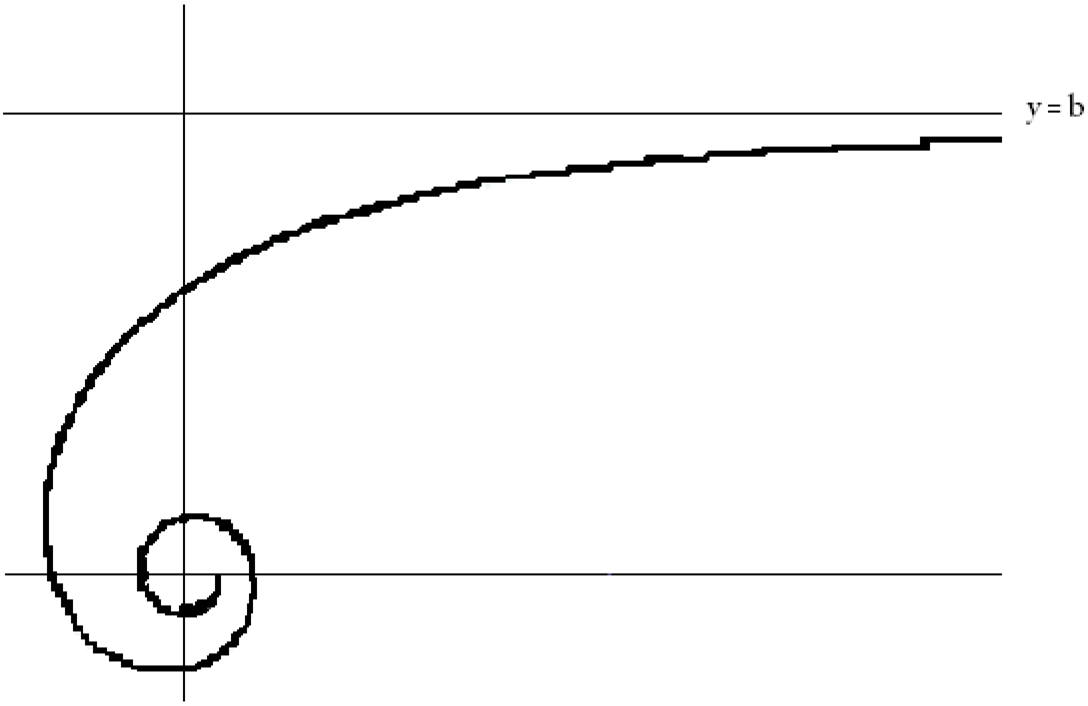
Hyperbolická spirála je inverzní k Archimédově spirále.

Rovnice pro hyperbolickou spirálu zní

$$r = \frac{b}{\varphi}$$

$b > 0, \quad \varphi \in R$

Hyperbolická spirála se zajímavě asymptoticky chová, pro $\varphi \to 0$ se body spirály blíží k přímce y = a, pro $\varphi \to \infty$ se délka průvodiče bodů blíží k nule.



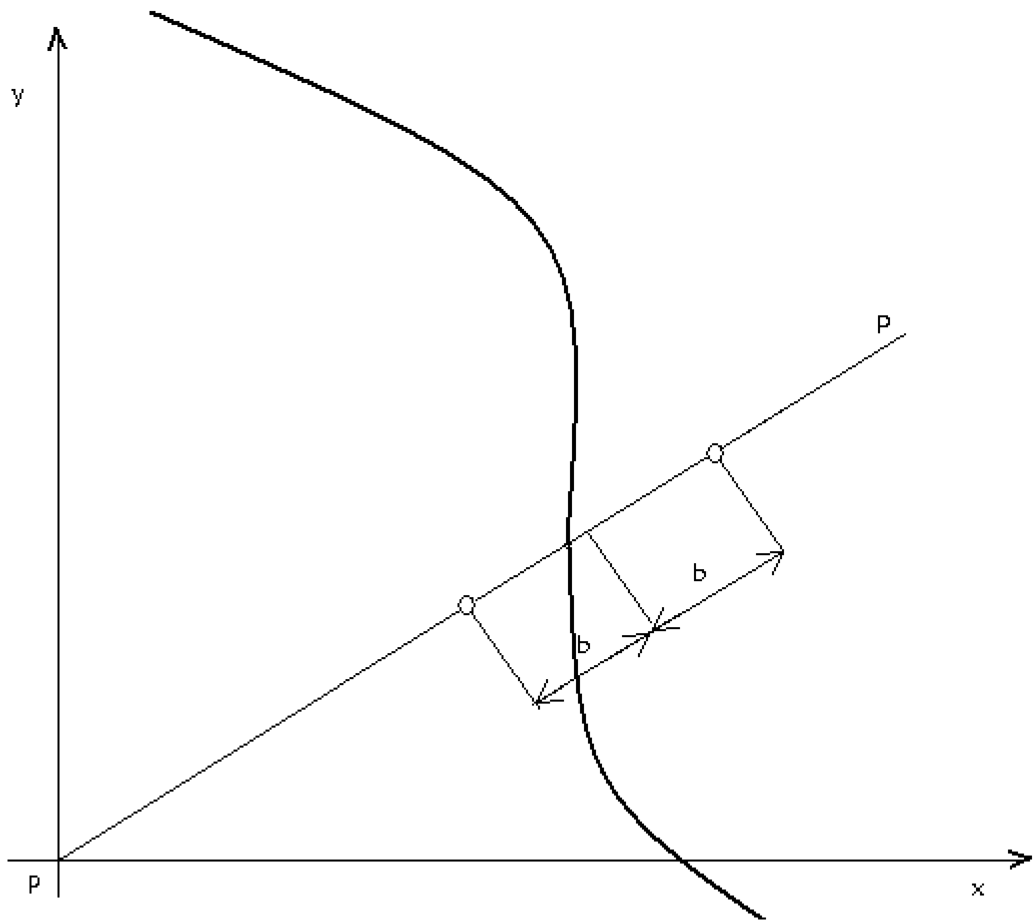
Obr. 15 Hyperbolická spirála

2.4. *Konchoidální křivky*

Když na každou přímku ***p*** svazku přímek o vrcholu ***P*** naneseme od průsečíku křivky ***k*** s přímkou ***p*** na obě strany úsečky o konstantní délce, získáme množinu bodů. Body, které jsou na kraji těchto úseček se nazývají ***přímá konchoida*** řídící křivky ***k*** pro vrchol ***P.***

Mezi konchoidální křivky patří:

- Nikomédova konchoida
- Pascalova závitnice



Obr. 16 Konchoida

2.4.1. Nikomédova konchoida

Nikomédova konchoida je přímá konchoida, která má řídící křivku přímku ***x* = *a*,** ***pro a*** > ***0***.

Rovnice Nikomédovy konchoidy v kartézských souřadnicích vypadá tak to

$$(x-a)^2(x^2+y^2)=b^2x^2$$

Rovnice v polárních souřadnicích lze vyjádřit tak to

$$r=\frac{a}{\cos\varphi}\pm b$$

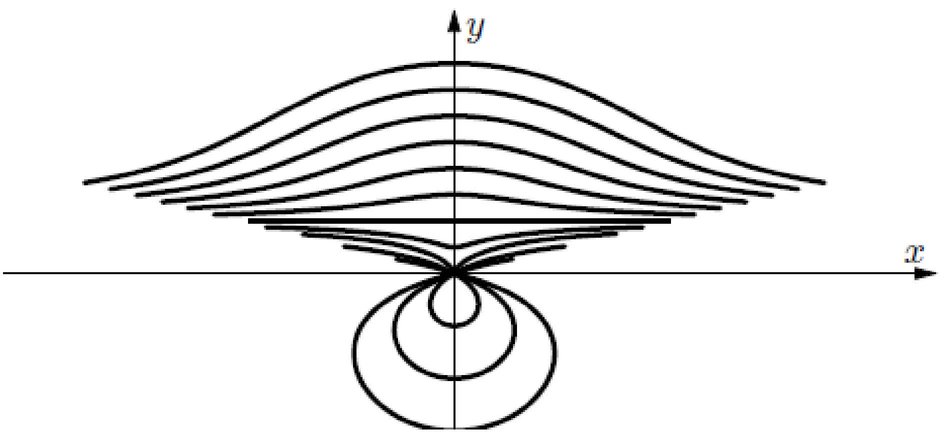
V bodech ***[a±b,0]*** má Nikomédova konchoida dva vrcholy a asymptotu ***x* = *a***. Pól ***P* =*[0,0]*** je pro ***b* < *a*** samostatným bodem, pro ***b* = *a*** je hrotem a pro ***b***>***a*** je dvojnásobným bodem, protože jedna z větví má uzel v pólu.

Parametrickou rovnici lze napsat ve tvaru

$$x=t+\frac{at}{\sqrt{b^2+t^2}}$$

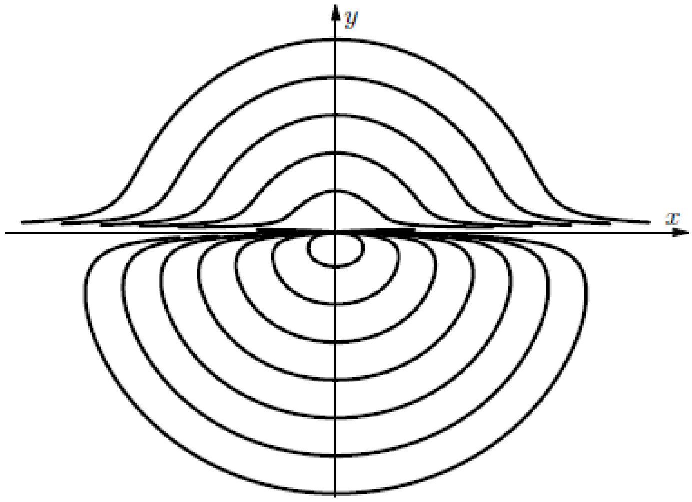
$$y=b+\frac{ab}{\sqrt{b^2+t^2}}$$

a, b jsou konstanty, ***t*** parametr podle nich můžeme Nikomédovu konchoidu znázornit různými tvary, například obrázek 17 a 18.



Obr. 17 Nikomédova konchoida

$a\in\langle -200;150\rangle, b=50, t\in\langle -20;20\rangle$



Obr. 18 Nikomédova konchoida

$a \in \langle -3500; 2500 \rangle, b = 50, t \in \langle -150; 150 \rangle$

2.4.2. Pascalova závitnice

Jestliže řídící křivkou je kružnice rovnice, na které leží vrchol ***V*** a parametr je ***b*** > ***0***, potom mluvíme o konchoidě kružnice neboli o ***Pascalovu závitnici***.

Pascalova závitnice jde vyjádřit rovnicí i v kartézských souřadnicích

$$(x^2 + y^2 - ax)^2 - b^2(x^2 + y^2) = 0$$

Rovnici v polárních souřadnicích můžeme napsat

$$\rho = a\cos\varphi \pm b$$

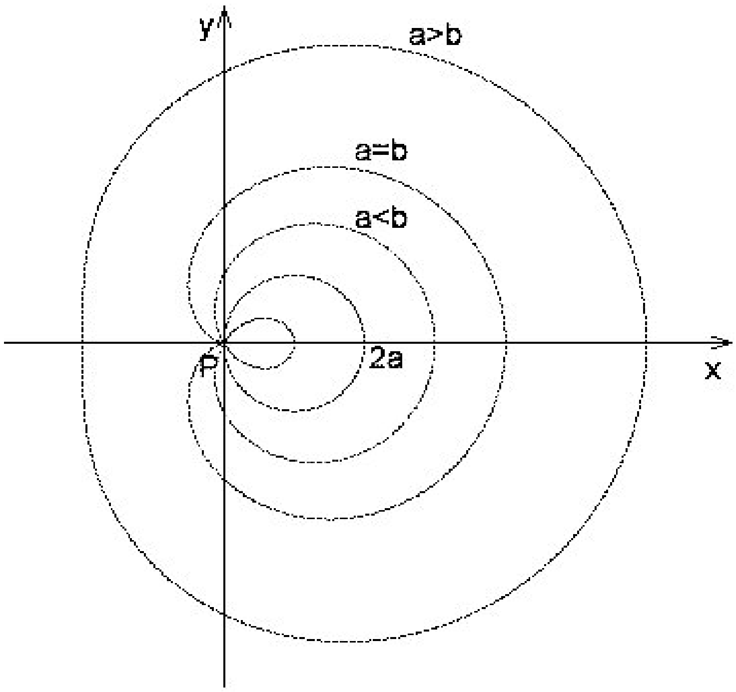
Parametrická rovnice pro Pascalovu závitnici je

$$x = a\cos t - b\cos 2t$$

$$y = a\sin t - b\sin 2t$$

Pokud se ***b=a*** potom se z Pascalovy závitnice stane kardioida.

Pokud je vrchol ***V =[0,0]*** Pascalovy závitnice potom je pro ***b < a*** uzlem, pro ***a=b*** vrcholem a pro ***b >a*** samotným bodem.



Obr. 19 Pascalova závitnice

3. Macromedia Flash

Počítačový program Macromedia Flash patří mezi komerční software, který umožňuje vytváření jednoduchých her, animací, webových stránek apod. Na Flashi je zajímavé to, že spojuje grafiku a programování, takže s ním může pracovat jak programátor, tak grafik. Flash má svůj programovací jazyk ActionScript, který slouží k rozvinutí všech možností interaktivní animace a vývoji robustních aplikací.

Bakalářská práce je vytvořena v Adobe Flash Professional CS3, který disponuje s programovacím jazykem ActionScript 3.

4. Implementace

Na následujících stránkách si ukážeme, jak jsem řešila problematiku animace cyklických, ekvidistantních, konchoidálních křivek a spirál.

Aplikace se skládá z :

- Prezentace_krivek.fla – nejvyšší instance pro animací křivek

a komponentů:

- CarkovanaCara.as - funkce pro vykreslení přerušované čáry, na základě definovaných parametrů
- KresliKrivku.as – funkce pro animování křivek obsažených v knihovně Krivka.as
- Krivka.as – knihovna funkcí křivek
- osy.as – funkce pro vykreslení os
- OvalnyCtvrec.as – funkce pro vykreslení oválných čtverců na základě definovaných parametrů
- Konfigurace.as – proměnné pro jednotlivé položky rolovacího menu výběru křivek a samotné parametry křivek a jejich nastavení pro vykreslení
- LoadHTMLCSS.as - funkce pro načtení HTML souborů s teorií do aplikace včetně podpory CSS stylu
- Pozadi.as – funkce pro vykreslení pozadí aplikace
- VyberMenu.as – funkce rolovací menu pro výběr křivek.

Ukážeme si pouze soubory Prezentace_krivek.fla, KresliKrivku.as a Krivka.as.

Funkce Prezentace_krivek.fla volá jednotlivé komponenty aplikace.

Funkce KresliKrivku.as provádí animaci křivky kreslené pomocí knihovny Krivka.as.

Funkce Krivka.as obsahuje funkce pro jednoduché vykreslení křivek.

Uživatelská a programátorská příručka je součástí přiloženého CD.

Prezentace_krivek.fla

System.useCodePage = true;

/* Import komponent */

import komponenty.OvalnyCtverec;

import komponenty.CarkovanaCara;

import komponenty.KresliKrivku;

import komponenty.Krivka;

import komponenty.Osy;

import komponenty.LoadHTMLCSS;

import komponenty.Pozadi;

import komponenty.Konfigurace;

import komponenty.VyberMenu;

stop();

/* #################### plocha ##################### */

var PozadiPre:Pozadi =new Pozadi();

addChild(PozadiPre);

/* ################ rolovaci meny ################## */

var RolovMenu:VyberMenu =new VyberMenu();

addChild(RolovMenu);

KresliKrivku.as

```
package komponenty

{

	import flash.display.MovieClip;

	import flash.display.Sprite;

	import flash.display.Graphics;

	import flash.events.Event;

	import flash.text.TextField;

public class KresliKrivku extends MovieClip

 {

	var krivkaKR:Krivka = new Krivka();

	var nakr_osy:Osy = new Osy(260, 330, 500, 500);

public function KresliKrivku()

 {

}

public function SmazKrivku()

 {

	krivkaKR.Smazat();

}

public function NactiKrivku(hranice:Object, krivkaParam:Object)

{

	var i:Number;

	var poziceX:Number = hranice.pozX;

	var poziceY:Number = hranice.pozY;

	var delkaX:Number = hranice.delX;

	var sirkaY:Number = hranice.sirY;
```

```
var graX:Number = krivkaParam.krivX;
var graY:Number = krivkaParam.krivY;
var str:Number = krivkaParam.krivStr;
var stp:Number = krivkaParam.krivStp;
var a:Number = krivkaParam.parmA;
var b:Number = krivkaParam.parmB;
var c:Number = krivkaParam.parmC;
var angD:Number = krivkaParam.uhel;


                addChild(krivkaKR);
                addChild(nakr_osy);
                trace("Krivka: "+krivkaParam.jmeno);
                krivkaKR.x = graX;
                krivkaKR.y = graY;
addEventListener(Event.ENTER_FRAME, onLoop, false, 0, true);
                i=str;
function onLoop(evt:Event):void
{
        krivkaKR.Smazat();
        if (krivkaParam.multipl==1)
{
        for (var j:Number = -krivkaParam.parmA; j<krivkaParam.parmA; j=j+10)
 {
        krivkaKR.Nakresli(krivkaParam.jmeno,j,b,c,str,i,angD,krivkaParam.evoluta);
}
}
        else
```

```
 {
krivkaKR.Nakresli(krivkaParam.jmeno,a,b,c,str,i,angD,krivkaParam.evoluta);
}
if (krivkaKR.Hranice()==1 || i>stp)
 {
removeEventListener(Event.ENTER_FRAME, onLoop);
        trace("stop");
        }
         else
         {
        i=i+angD;
                }
        }
}
    }
}
```

Krivka.as

```
package komponenty

{

	import flash.display.MovieClip;

	import flash.text.TextField;

	import flash.display.Sprite;

	import flash.display.Graphics;

	import flash.events.Event;

public class Krivka extends MovieClip

{

	var krivka:Sprite = new Sprite;

	var hraX, hraY, hraDel, hraSir:Number;

	var hranice:Number = 0;

	var MimoHranice:Number = 0;

	var x_pos:Number = 0;

	var y_pos:Number = 0;

	var x_old:Number = 0;

	var y_old:Number = 0;

public function Krivka()

{

	addChild(krivka);

}

public function Nakresli(k:String, a:Number, b:Number, c:Number, st:Number, kon:Number,
angD:Number, evol:Number )

{

	var frst:Number = 0;
```

```
            var degToRad = Math.PI/180;

            var ang:Number = 0;

for(var t:Number = st; t <= kon; t=t+angD)

            {

            ang = t*degToRad;

            switch(k)

            {

            case "nikodem":

            x_pos = t+a*t/Math.pow(b*b+t*t,1/2);

            y_pos = b+a*b/Math.pow(b*b+t*t,1/2);

            break;

            case "pascal":

            x_pos = a*Math.cos(ang)-b*Math.cos(2*ang);

            y_pos = a*Math.sin(ang)-b*Math.sin(2*ang);

            break;

            case "archimed":

            x_pos = a*Math.cos(ang)-b*Math.cos(2*ang);

            y_pos = a*Math.sin(ang)-b*Math.sin(2*ang);

            break;

            case "cykloida":

            x_pos=a*ang+b*Math.sin(ang);

            y_pos=a-b*Math.cos(ang);

            break;

            case "epicykloida":

            x_pos=(a+b)*Math.cos(ang)-b*Math.cos(ang*(a+b)/b);

            y_pos=(a+b)*Math.sin(ang)-b*Math.sin(ang*(a+b)/b);
```

```
break;
case "epicykloida_zp":
x_pos=(a+b)*Math.cos(ang)-c*Math.cos(ang*(a+b)/b);
y_pos=(a+b)*Math.sin(ang)-c*Math.sin(ang*(a+b)/b);
break;
case "hypocykloida":
x_pos=(a-b)*Math.cos(ang)+b*Math.cos(ang*(a-b)/b);
y_pos=(a-b)*Math.sin(ang)-b*Math.sin(ang*(a-b)/b);
break;
case "hypocykloida_zp":
x_pos=(a-b)*Math.cos(ang)+c*Math.cos(ang*(a-b)/b);
y_pos=(a-b)*Math.sin(ang)-c*Math.sin(ang*(a-b)/b);
break;
case "evolventa":
x_pos=(a+b)*Math.cos(ang)+a*ang*Math.sin(ang);
y_pos=(a+b)*Math.sin(ang)-a*ang*Math.cos(ang);
break;
case "archimedsp":
x_pos=a*ang*Math.cos(ang);
y_pos=a*ang*Math.sin(ang);
break;
case "logaritmsp":
x_pos=b*Math.pow(a,ang)*Math.cos(ang);
y_pos=b*Math.pow(a,ang)*Math.sin(ang);
break;
case "fermatsp":
```

```
x_pos=b*Math.sqrt(ang*a)*Math.cos(ang);
y_pos=b*Math.sqrt(ang*a)*Math.sin(ang);
break;
case "hyperbolsp":
x_pos=a/ang*Math.cos(ang);
y_pos=a/ang*Math.sin(ang);
break;
case "kruznice":
x_pos = a*Math.sin(ang);
y_pos = a*Math.cos(ang);
break;
}
krivka.graphics.lineStyle(2,0x0000FF);
        if (frst==0)
{
        krivka.graphics.moveTo(x_pos,y_pos); // štětec
                x_old=x_pos;
                y_old=y_pos;
                frst=1;
}
        krivka.graphics.lineTo(x_pos,y_pos); // vykreslení
                if(evol>0)
                {
                Evoluta(evol);
                                }
                        if (hranice==1 && Hranice()==1)
```

```
{
        break;
        }
}
        switch(k)
        {
        case "cykloida":
        krivka.graphics.lineStyle(1,0x000000,10);
        krivka.graphics.lineTo(ang*a,a);
        krivka.graphics.lineTo(ang*a,a+b);
        krivka.graphics.drawCircle(ang*a,a,b);
        krivka.graphics.drawCircle(ang*a,a,a);
        break;
        case "epicykloida":
        krivka.graphics.lineStyle(1,0x000000,10);
        krivka.graphics.drawCircle(0,0,a);
        krivka.graphics.lineStyle(1,0x000000,10);
        krivka.graphics.drawCircle((a+b)*Math.cos(ang),(a+b)*Math.sin(ang),b);
        break;
        case "epicykloida_zp":
        krivka.graphics.lineStyle(1,0x000000,10);
        krivka.graphics.drawCircle(0,0,a);
        krivka.graphics.lineStyle(1,0x000000,10);
        krivka.graphics.drawCircle((a+b)*Math.cos(ang),(a+b)*Math.sin(ang),b);
        krivka.graphics.drawCircle((a+b)*Math.cos(ang),(a+b)*Math.sin(ang),c);
        break;
```

```
case "hypocykloida":
krivka.graphics.lineStyle(1,0x000000,10);
krivka.graphics.drawCircle(0,0,a);
krivka.graphics.lineStyle(1,0x000000,10);
krivka.graphics.drawCircle((a-b)*Math.cos(ang),(a-b)*Math.sin(ang),b);
break;
case "hypocykloida_zp":
krivka.graphics.lineStyle(1,0x000000,10);
krivka.graphics.drawCircle(0,0,a);
krivka.graphics.lineStyle(1,0x000000,10);
krivka.graphics.drawCircle((a-b)*Math.cos(ang),(a-b)*Math.sin(ang),b);
krivka.graphics.drawCircle((a-b)*Math.cos(ang),(a-b)*Math.sin(ang),c);
break;
case "evolventa":
krivka.graphics.lineStyle(1,0x000000,10);
krivka.graphics.drawCircle(0,0,a);
krivka.graphics.drawCircle(0,0,b+a);
krivka.graphics.lineStyle(1,0x000000,10);
krivka.graphics.moveTo(x_pos,y_pos);
krivka.graphics.lineTo((a+b)*Math.cos(ang),(a+b)*Math.sin(ang));
break;
case "archimedsp":
krivka.graphics.lineStyle(1,0x000000,10);
krivka.graphics.moveTo(x_pos,y_pos);
krivka.graphics.lineTo(0,0);
break;
```

```
case "logaritmsp":
krivka.graphics.lineStyle(1,0x000000,10);
krivka.graphics.moveTo(x_pos,y_pos);
krivka.graphics.lineTo(0,0);
break;
case "fermatsp":
krivka.graphics.lineStyle(1,0x000000,10);
krivka.graphics.moveTo(x_pos,y_pos);
krivka.graphics.lineTo(0,0);
break;
case "hyperbolsp":
krivka.graphics.lineStyle(1,0x000000,10);
krivka.graphics.moveTo(x_pos,y_pos);
krivka.graphics.lineTo(0,0);
break;
}
krivka.graphics.lineStyle(1,0x00FF00,10);
krivka.graphics.drawCircle(x_pos,y_pos,2);
}
public function Evoluta(delka:Number)
 {
        var x_vec:Number = 0;
        var y_vec:Number = 0;
        var x_jvec:Number = 0;
        var y_jvec:Number = 0;
        x_vec = y_pos-y_old;
```

```
                y_vec = -(x_pos-x_old);
                x_jvec = x_vec/(Math.pow(x_vec*x_vec+y_vec*y_vec,1/2));
                y_jvec = y_vec/(Math.pow(x_vec*x_vec+y_vec*y_vec,1/2));
                krivka.graphics.lineStyle(1,0xa00000,20);
krivka.graphics.moveTo((x_pos+x_old)/2+delka*x_jvec,(y_pos+y_old)/2+delka*y_jvec);
krivka.graphics.lineTo((x_pos+x_old)/2-delka*x_jvec,(y_pos+y_old)/2-delka*y_jvec);
                krivka.graphics.moveTo(x_old,y_old);
                        x_old=x_pos;
                        y_old=y_pos;
                        }
        public function NastavHranice(hraniceD:Number, hraniceS:Number)
                {
                    hraX=-hraniceD/2;
                    hraY=-hraniceS/2;
                    hraDel=hraniceD/2;
                    hraSir=hraniceD/2;
                    hranice = 1;
        trace("Hranice nastavena"+hraX+","+hraY+":"+hraDel+","+hraSir);
                }
                public function Hranice()
                {
                if ((x_pos)<hraX || (y_pos)<hraY || (x_pos)>hraDel || (y_pos)>hraSir)
                {
                MimoHranice = 1;
trace("Mimo hranice x:"+x_pos+" y:"+y_pos+" > "+hraX+","+hraY+":"+hraDel+","+hraSir);
```

```
}
else
{
MimoHranice = 0;
}
return MimoHranice;
}
public function Smazat()
 {
krivka.graphics.clear();
            }
}
}
```

5. Závěr

Cílem bakalářské práce bylo vytvořit výukové programy zaměřené na problematiku řešení geometrických úloh spirál a cyklických, ekvidistantních, konchoidálních křivek. Důraz při tvorbě aplikací jsem kladla především na vzhled a srozumitelnost pro uživatele.

Pro práci jsem si proto vybrala Macromedia Flash doplněný programovacím jazykem ActionScript 3, který díky svým rozsáhlým možnostem nabízí prostředí pro tvorbu animací. Jednotlivé křivky jsou doplněny teorii, samostatnou animací popisované křivky a krátkým popisem.

6. Literatura

[1] Kerman P. *Action Script ve Flashi Podrobná příručka* Computer Press Praha 2002

[2] Schneider Z. *Macromedia Flash jednoduše* Computer Press Praha 2002

[3] Fotr J., Schneider Z. *Flash 5 pro grafiky a tvůrce webů* Computer Press Praha 2000

[4] Schneider Z., Jirka L. *Vytváříme hry ve Flashi* Grada Publishing Praha 2004

[5] Ježek F. *Diferenciální geometrie* Západočeská univerzita v Plzni 2005

[6] Procházková J. *Cyklické pohyby s využitím softwaru Cabri a Maple* Masarykova univerzita v Brně 2004

[7] Říhová H. *Křivky* 2006

[8] http://barvy.gene.cz/ba-hex.html

[9] http://cs.wikipedia.org/wiki/Cyklick%C3%A1_k%C5%99ivka

[10] http://cs.wikipedia.org/wiki/Cykloida

[11] http://cs.wikipedia.org/wiki/Epicykloida

[12] http://cs.wikipedia.org/wiki/Hypocykloida

[13] http://cs.wikipedia.org/wiki/Spir%C3%A1la

[14] http://cs.wikipedia.org/wiki/Konchoida

[15] http://cs.wikipedia.org/wiki/Evoluta

[16] http://cs.wikipedia.org/wiki/Evolventa

[17] http://flash.jakpsatweb.cz/index.php?page=seznameni

[18] http://fo.cuni.cz/texty/matematika/mkrivek.pdf

[19] http://www.math.muni.cz/~pospisl/dgkp/evoluta.html

[20] http://www.pf.jcu.cz/cabri/temata/CYKLOIDY/index.html#evolutaepi

7. Seznam příloh

CD

- kopie tohoto dokumentu
- programátorská dokumentace
- uživatelská dokumentace
- animace